Monocular Depth Estimation

Christian Chang, Justin Hall, Lyndsey Mandelare
School of Robotics Engineering
Worcester Polytechnic Institute
Worcester, MA 01609
Email: cmchang@wpi.edu, jrhall@wpi.edu, lkmandelare@wpi.edu

***Abstract*—Depth estimation in images is a core capability of computer vision-enabled automated driving. Monocular depth estimation represents the opportunity to decrease the cost of this capability significantly by using Deep Learning. This investigation will aim to provide a monocular solution to depth estimation that maximizes accuracy.**

I. Introduction

The field of automated driving has a multifaceted problem base, much of which has solutions based in Artificial Intelligence. Artificial Intelligence is a subset of machine learning, which aims to teach computers, in human-like manners, to solve problems. This can be applied to nearly every part of automated driving as it is a task that computers are taking over from humans. However, the field of computer vision uses artificial intelligence extensively for such tasks as object detection and tracking.

Another task within computer vision in an automated driving context is depth estimation. Depth estimation is the process of determining distances from objects of interest in a scene. Traditionally, this is accomplished with stereo cameras and a Li-DAR sensor. This allows for 3D mapping and object detection, but is expensive in both money and processing power. Recently, there has been an emergence of research into monocular depth estimation, which can create depth estimations from a single image. This method is accomplished by using traditionally collected depth estimation data sets to train the Deep Learning algorithm. Some of the most successful algorithms also use geometric constraints to increase the accuracy of the depth estimations.

II. Literature Review

Recent research into the depth estimation from 2D images and monocular cameras has yielded improvements in depth estimation and uses a variety of methods. [2] used recurrent neural networks and an ensemble network architecture called SU-Nets. Their research consisted of two network architectures connected in series which they called a W-Net. This represented the simplest series network architecture conceivable for the purpose of their experiment. The result was reduced training and validation losses thru the use of the W-Net Connected architecture. Depth predictions were best using the W-Net Connected architecture proposed in the research.

The survey of monocular depth estimation methods finds there are a number of deep learning methods that use convolutional neural networks (CNN) [1]. The deep learning networks in the survey use various levels of supervision to accomplish depth estimation. The survey concluded that, of the highest performing deep learning networks, "DeepV2D marginally outperforms BTS and the rest of the methods" in accuracy and depth range on the KITTI and NYUD-v2 datasets [1]. Per [1], DeepV2D is a CNN utilizing supervised learning while BTS utilizes encoder-decoder architecure and supervised learning[1].

A. MonoDepth2

In an effort to reduce the need on difficult to obtain ground truth data for monocular depth estimation, the creators of Monodepth 2 sought to create a self-supervised model. The Monodepth 2 model for monocular depth estimation uses three innovations: minimum reprojection loss, full-resolution multi-scale sampling method that reduces visual artifacts and auto-masking loss. This model trains using monocular video or pairs of stereo images. Training can be done with both as well.

B. DeepV2D

The primary publication on the DeepV2D algorithm is the Teed and Deng paper published as a conference paper at ICLR 2020. This paper expanded upon the principle that by combining image data with a mathematical model of how images are formed, more accurate predictions of depths could be made from single images. This was acheived by combining two processes: camera motion estimation and depth estimation, which feed each other's predictions. The camera motion model uses a new algorithm called Flow-SE3, which creates a dense 2D correspondence array between two frames. The Depth estimation model uses Multi-view Stereo reconstruction and trainable architectures for feature extraction and matching. This was a unique contribution because although previous models combined camera motion models and depth estimation, this was the first algorithm to allow them to feed into each other and alternate between tasks as needed. [3]

C. ManyDepth

ManyDepth is a relatively new monocular depth estimation model that was developed by a team at Naintic Labs. Niantic

Labs expanded upon existing types of self-supervies depth estimation models in three main ways. First, their model utilizes sequential frames of a video as a type of stereo depth estimation while in motion as well as uses a single frame estimation during initialization and while the camera is static. Second, they utilize the single frame depth estimation to improve artifacts in images caused by moving objects. Finally, they have developed an adaptive cost volume that learns from data rather then a set of parameters. [8]

The model can be trained with a sequence of 3 frames, either past-current-future (I_{t-1},I_t,I_{t+1}) or past-past-current (I_{t-2},I_{t-1},I_t). At test time only the current frame and the previous frame is required for depth estimation. Base on Niantic Labs findings, ManyDepth out performed other self-supervised models in almost every category.

D. Wavelet Monodepth

The primary publication in regards to Wavelet Monodepth is Single Image Depth Prediction with Wavelet Decomposition by a team at Niantic. This method uses the framework for monocular depth estimation established by the monodepth algorithm, but changes how depth is represented within the system to increase efficiency of the algorithm. The depth representation is done using wavelet coefficients. Wavelet coefficients are used to represent depth in an image. Wavelet coefficients are the result of wavelet decomposition of a section of an image. These coefficients can then be used to reconstruct the larger image from a smaller matrix of coefficients. This enables compression and denoising processes for many use cases. Some depth prediction algorithms use wavelets to compress the images instead of resizing or cropping images for analysis, since wavelets produce less loss than these methods. However, this algorithm goes further by directly predicting from the wavelet coefficients, rather than the image matrix. Additionally, this implementation of wavelet decomposition is completely self-supervised, unlike previous algorithms that use wavelets, and has variable sparsity by changing the percentage of pixels that wavelet coefficients were computed for. This reduces the number of multiply-add operations within the decoder by up to a factor of 2 on dense networks with coefficients from 10 percent of pixels. [9]

III. Problem Description

3D sensors and cameras can range in cost from $250 to $4000, compared to a simple 2D camera that ranges from $10 to $50. This may not seem like a lot in the overall cost of an autonomous vehicle (AV), but this could mean significant cost savings once AV becomes the norm. This does not include potential savings on industrial robots. The main issue is a single camera only has a 2D image to estimate depth, so standard stereo camera methods cannot be used.

For AVs, it is also vitally important to have fast and reliable depth information for obstacle avoidance and detection. Poor depth estimation can lead to vehicles responding to obstacles too late or too soon, which could end in a collision and potential loss of life. Slow time to estimate depth information could lead to a similar scenario. In this paper we will provide a solution for monocular depth estimation for obstacle avoidance.

IV. Methods

This paper aims to survey the strengths and weaknesses of three monocular depth estimation techniques to determine the a method of combining their strengths into a more effective algorithm for autonomous vehicle navigation. To this end, we will be investigating the MonoDepth, Wavelet MonoDepth, and ManyDepth algorithms. These algorithms will each be tested against the KITTI data set, chosen for its relevance to autonomous driving.

A. Training Environment

Multiple environments were used to test the different datasets. ManyDepth utilized a Nvidia NVS510 GPU on a home PC, as well as a Tesla P100 within Google Colab. The pretrained KITTIHR model provided by the developers were used during testing.

Monodepth 2 was tested on pretrained models provided via the developers Github. The Monodepth 2 model used was the mono+stereo_1024x320. This model was trained in monocular and stereo images at the highest resolution used by Monodepth 2, which is 1024x320.

B. Data Set

For the model comparison we used five images from a single clip of the KITTI data-set [7]. The clip chosen was 2011_09_28_drive_0001_sync. This clip had a good variety of clustered objects, moving objects, and a right hand turn at a light. This allowed us to give a few different scenarios to which model performs the best over all situations.

Ground truth data from the same data set was also utilized in this project. Results from each method were compared to the ground truth and then compared to each other.

V. Results

A. ManyDepth

The output depth maps for the ManyDepth model look very good visually as can be seen in Figure 1. Unfortunately, we found that the numbers associate with these depths were reversed. The brighter yellow color associates with a big number and the dark color associates with a small number.

Fig. 1: ManyDepth Depth Map for Image 5

This is the opposite of what the output of our ground truth image show as can be seen in Figure 2. To account for this we inverted the distances of the ManyDepth output by taking the absolute value of the output array minus one, since the scale

of the array is from zero to one. Also, in the ground truth there is a dark area on the top and right side of the image. When interpolating the ground truth image from the raw ground truth data there was not enough information in those sections to provide good depth information. These section are left at zero, which is obviously incorrect. This may account for the higher then expected root mean squared error (RMSE) values discussed later.

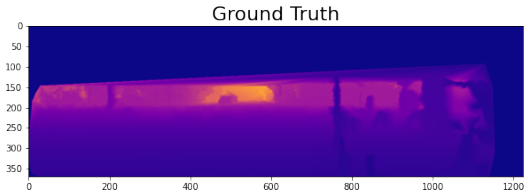

Fig. 2: ManyDepth Ground Truth Depth Map for Image 5

Figure 3 shows the inverted output that matches the ground truth, but it is set to it own custom scale. All other images aside from the 'Predicted' images will have the depth map color scale scaled from zero to 100. Figure 4 also was scaled up by a factor of 54 to bring the depth distances to the same scale as the ground truth.

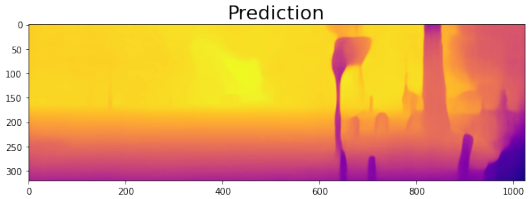

Fig. 3: ManyDepth Inverted Output Results for Image 5

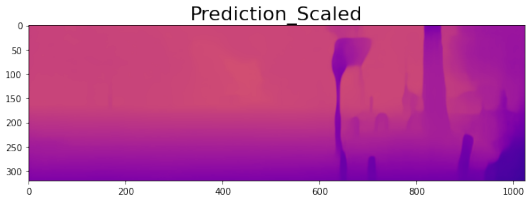

Fig. 4: ManyDepth Scaled Output Results for Image 5

After the output data has been scaled we compared the scaled output array to the ground truth array by taking the absolute value of the difference of the two arrays to give the error array shown in Figure 5. The entire sequence of images is shown in Figure 6 is for the first test image. This also includes the color scale used for all images except the 'Prediction'.

The rest of the test results for Many depth are in the Appendix. The results are full sequences similar to Figure 6, but with out the reference color bars. All in all the results are good, but with the absence of crucial information on the depth map we were not able to recreate the same RMSE values of 4 produced by the original ManyDepth tests.

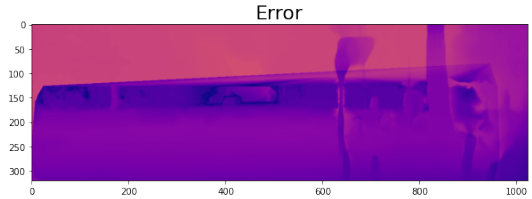

Fig. 5: ManyDepth Error Results for Image 5

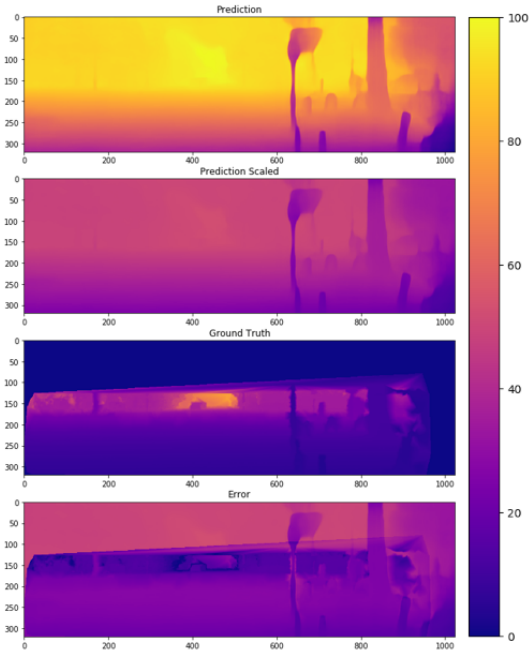

Fig. 6: ManyDepth Output Results for Image 5

B. MonoDepth2

Test images 5 and 100 are shown in Figures 7 and 9. Corresponding Monodepth2 depth predictions are shown in Figures 8 and 10. Error maps were created in the same fashion as for Manydepth. Corresponding full sequence and error maps are shown in Figures 25 and 29 in the appendix.

Fig. 7: Test Image 5

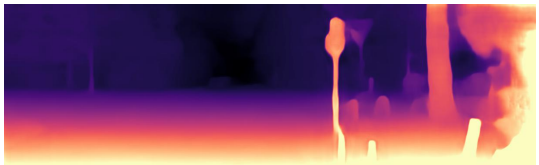

Fig. 8: Monodepth 2 Prediction for Image 5

Trials for Monodepth depth predictions were generally successful. Visual inspection shows that objects a human would interpret as nearby or far-away are well differentiated in the prediction heatmaps.

Fig. 9: Test Image 100

Fig. 10: Monodepth 2 Prediction for Image 100

Even in areas that are shaded or have high glare, objects depths are well differentiated. One area of concern for the Monodepth is a glaring error in the depth map prediction. Take for example the yield sign shown in Image 5. It is plain to see this sign is attached to the traffic light pole, therefore their depths should be predicted to be similar. Yet the depth map in Figure 8 shows the yield sign to be significantly further from the vehicle than the traffic light. This is a cause for concern.

C. Wavelet MonoDepth

Wavelet monodpeth produces the following depth maps for the test images when using the dense encoder/decoder network called Resnet with 50 layers on 1024x320 KITTI images. Figure 11 shows the depth prediction of wavelet monodepth for image 5.

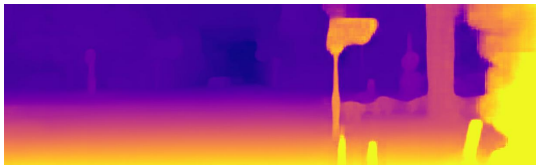

Fig. 11: Wavelet Monodepth Output Results for Image 5

Much like the Many Depth data, the values are inverted from the ground truth scale, so the images were manipulated in the same way as the many depth predictions to enable comparison with the ground truth data. Figure 12 shows the full results for image 5.

Also of interest are the full results for images 80 and 100. Figure 13 shows the full results for image 80. Figure 14 shows the full results for image 100. These results show a strange artifact in the prediction of cars very near the camera: it predicts that the car hood is further away from the camera than the middle of the car.

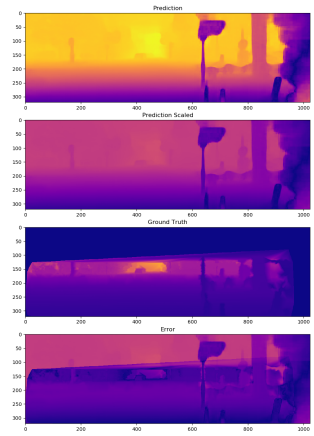

Fig. 12: Wavelet Monodepth Full Results for Image 5

Fig. 13: Wavelet Monodepth Output Results for Image 80

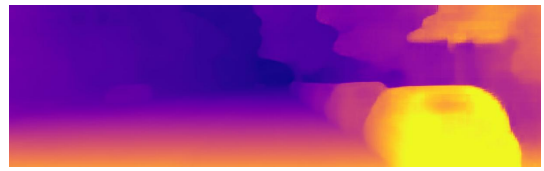

Fig. 14: Wavelet Monodepth Output Results for Image 100

D. Comparison

Table I shows the RMSE of each method for each test image. Monodepth2 performed the best in our depth prediction test, with the lowest RMSE on every image. This may be because Monodepth2 functions purely on monocular data, while Many Depth performs best with a video feed, getting more accurate as more frames are seen. Meanwhile, Wavelet Monodepth is designed to reduce prediction times and the computational load of the Monodepth algorithm, rather than increase the accuracy.

Root Mean Square Error					
Method	img5	img48	img64	img80	img100
Many Depth	33.025	35.323	31.429	31.689	31.678
Wavelet Monodepth	27.54	30.02	25.07	28.30	26.40
Monodepth2	16.835	16.187	18.547	17.420	16.907

TABLE I: All Methods RMSE Values for each Image

VI. Issues

The original approach to this paper was to analyze the DeepV2D algorithm put forward by Princeton and attempt to rectify the main drawback in the algorithm: memory usage. However, the team faced challenges getting the code base to run on our systems due to its use of legacy versions of tensorflow and cuda. Initially, our answer to this was to remove the emphasis on the V2D algorithm by introducing other algorithms and comparing them to V2D. This would have allowed us to see how others had improved upon the drawbacks of this algorithm. However, as the semester came to a close, the team was able to get these other algorithms functioning and producing results, but there was still no progress with the V2D algorithm. Therefore, we had to cut out the V2D algorithm from our report entirely and switch to the Wavelet MonoDepth algorithm. This new algorithm was chosen because of the similarity to the other algorithms in test and the quality of the documentation.

Another initial objective was to compare speed of each of the three algorithms to identify which was the most efficient. Based on the team aspect of this project and the differing computer make-ups this was not a realistic option. Especially with the difficulties of setting up the environments to get the models running. The use of Google Colab was also attempted, but some of the python versions and module installations were not compatible with this environment either.

Finally, we had issues understanding the data outputted by the models. It took a bit to understand that the numbers associated to the depth estimations were inverted when compared to the ground truth, as well as understanding the scaling factor to bring the normalized values to ones that correlate to a real world depth.

VII. Conclusion

This is not the outcome that we were expecting based on the other tests run in the ManyDepth paper. The RMSE values that our test outputted was significantly higher that those presented by each algorithms paper. This may be due to the large areas of no data that our ground truths presented. Regardless of this, it is very clear visually as well as based on the RMSE values that, for our test, the MonoDepth2 algorithm performed the best.

References

[1] Khan, Faisal and Salahuddin, Saqib and Javidnia, Hossein. (2020). Deep Learning-Based Monocular Depth Estimation Methods-A State-of-the-Art Review.

[2] Cantrell, K., Miller, C. and Morato, C. Practical Depth Estimation with Image Segmentation and Serial U-Nets.(2020). Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2020), pages 406-414 ISBN: 978-989-758-419-0

[3] Zachary Teed and Jia Deng DeepV2D: Video To Depth with Differentiable Structure From Motion.(2020). Proceedings of ICLR 2020.

[4] Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., amp; Firman, M. (2021). The temporal opportunist: Self-supervised multi-frame monocular depth. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr46437.2021.00122

[5] Cl´ement Godard, C., Aodha, O., Firman, M., Brostow, G. (2019). Digging Into Self-Supervised Monocular Depth Estimation. The International Conference on Computer Vision (ICCV). https://arxiv.org/abs/1806.01260

[6] Ramamonjisoa, Michael and Michael Firman and Jamie Watson and Vincent Lepetit and Daniyar Turmukhambetov. (2021) Single Image Depth Prediction with Wavelet Decomposition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://arxiv.org/abs/2106.02022

[7] KITTI Single Depth Evaluation Server. http://www.cvlibs.net/datasets/kitti/eval depth.php? benchmark=depth prediction. 2017.

[8] Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., amp; Firman, M. (2021). ManyDepth https://github.com/nianticlabs/manydepth

[9] Ramamonjisoa, Michael and Michael Firman and Jamie Watson and Vincent Lepetit and Daniyar Turmukhambetov. (2021) wavelet-monodepth https://github.com/nianticlabs/wavelet-monodepth

[10] Cl´ement Godard, C., Aodha, O., Firman, M., Brostow, G. (2019). monodepth2 https://github.com/nianticlabs/monodepth2

APPENDIX

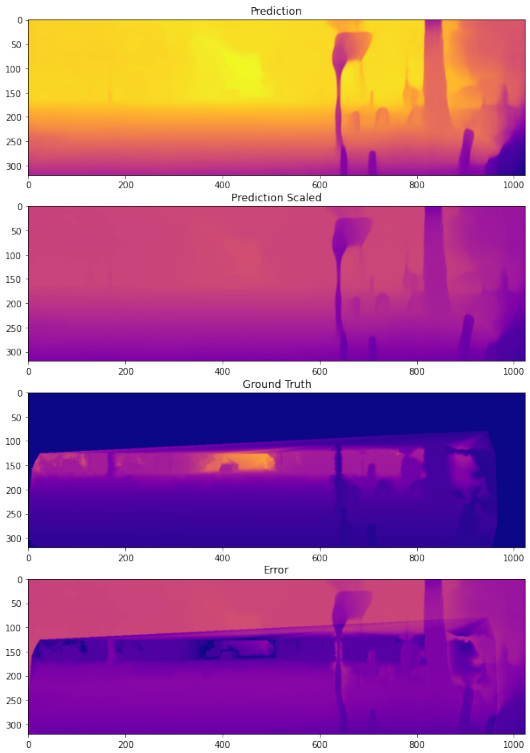

Fig. 15: ManyDepth Output Results for Image 5

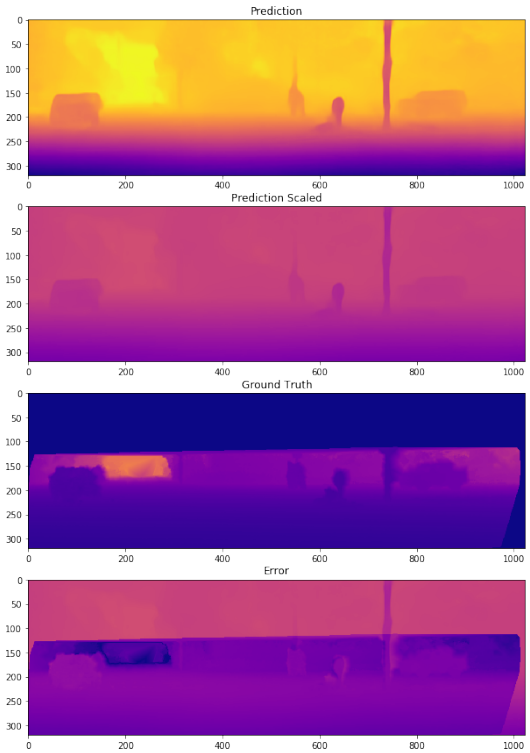

Fig. 16: ManyDepth Output Results for Image 48

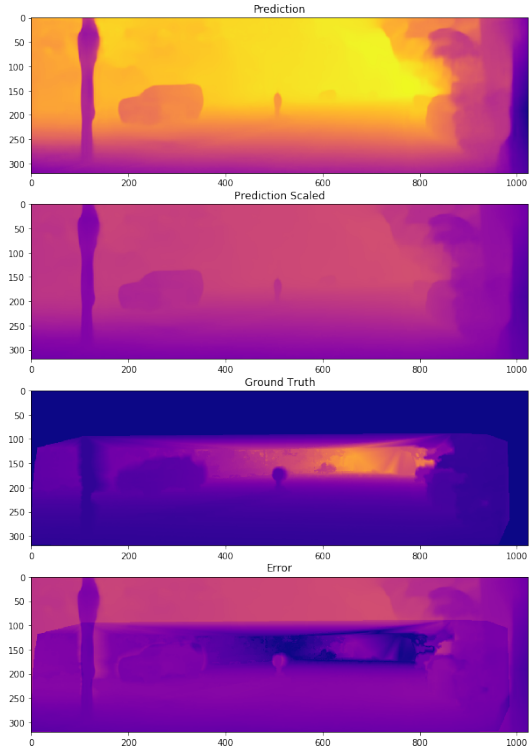

Fig. 17: ManyDepth Output Results for Image 64

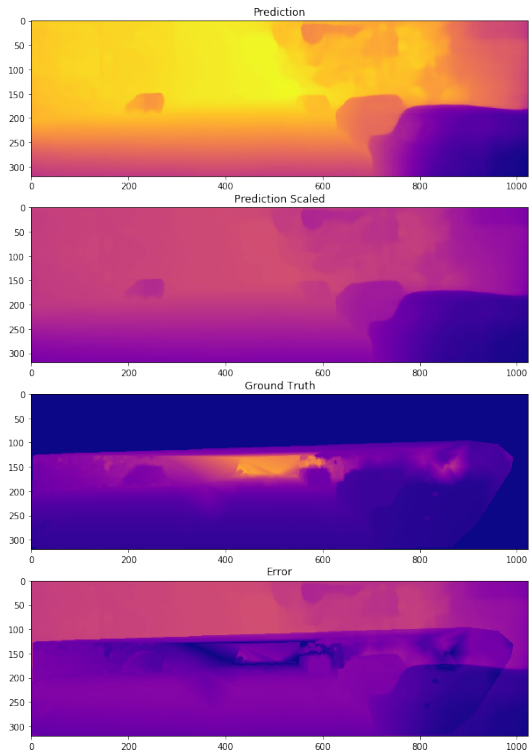

Fig. 18: ManyDepth Output Results for Image 80

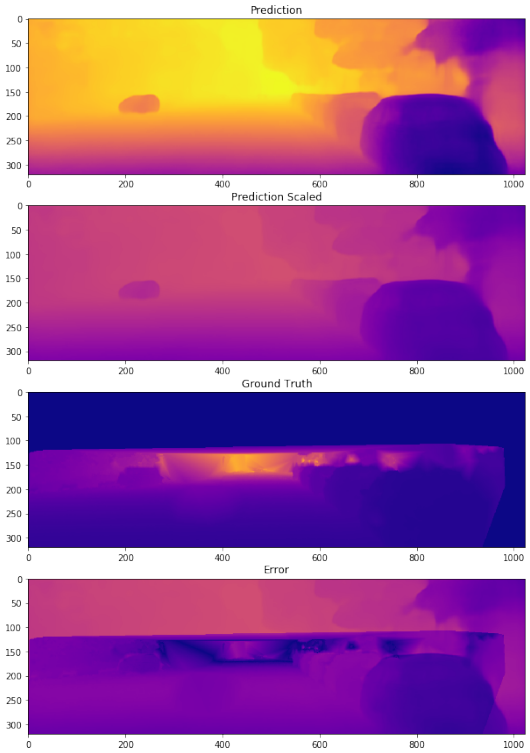

Fig. 19: ManyDepth Output Results for Image 100

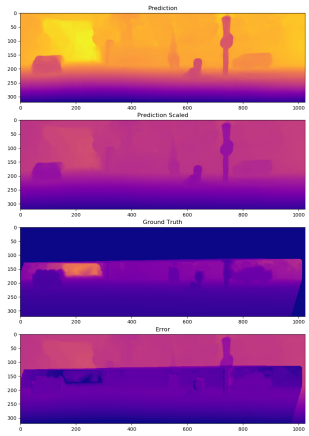

Fig. 21: Wavelet Monodepth Full Results for Image 48

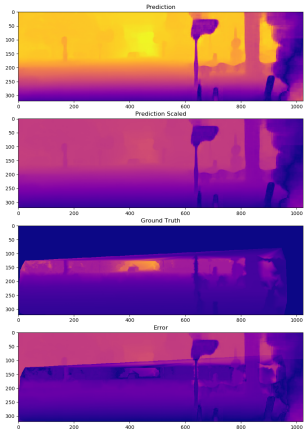

Fig. 20: Wavelet Monodepth Full Results for Image 5

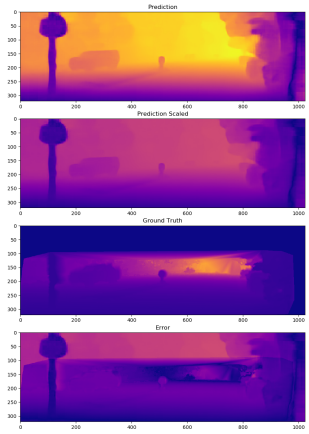

Fig. 22: Wavelet Monodepth Full Results for Image 64

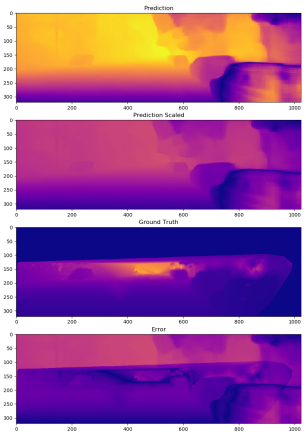

Fig. 23: Wavelet Monodepth Full Results for Image 80

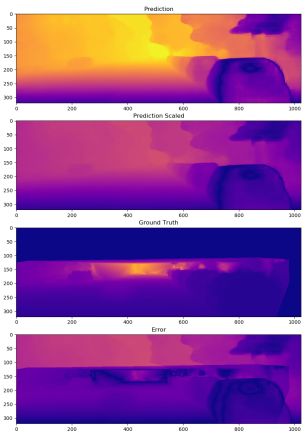

Fig. 24: Wavelet Monodepth Full Results for Image 100

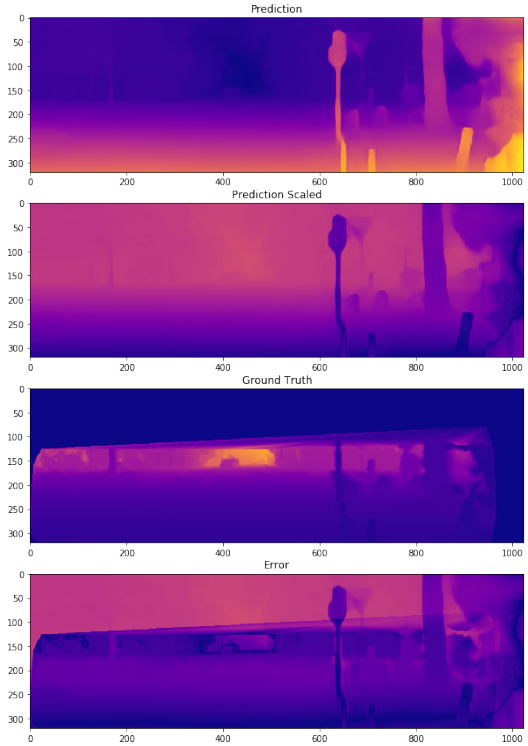

Fig. 25: Monodepth 2 Full Results for Image 5

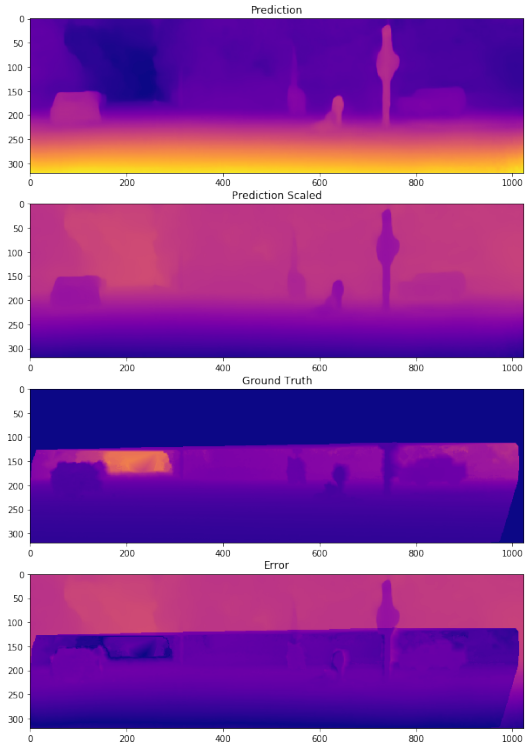

Fig. 26: Monodepth 2 Full Results for Image 48

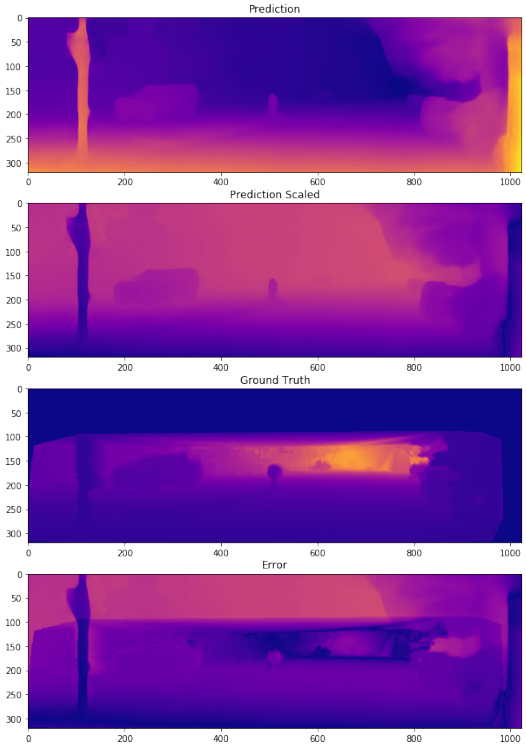

Fig. 27: Monodepth 2 Full Results for Image 64

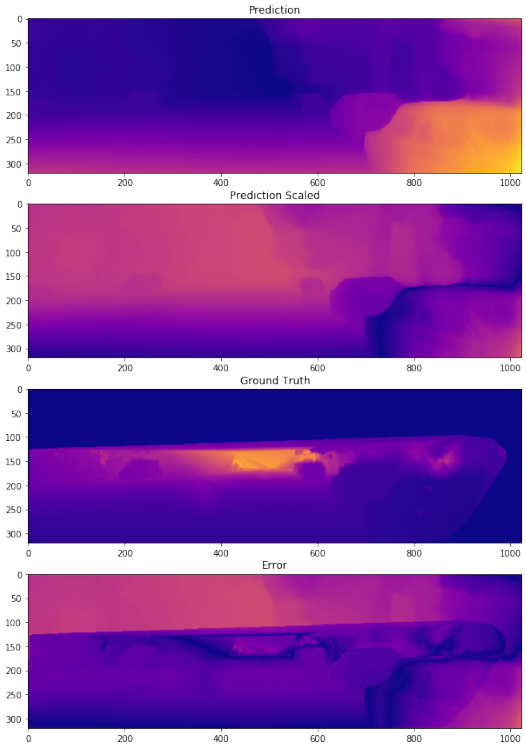

Fig. 28: Monodepth 2 Full Results for Image 80

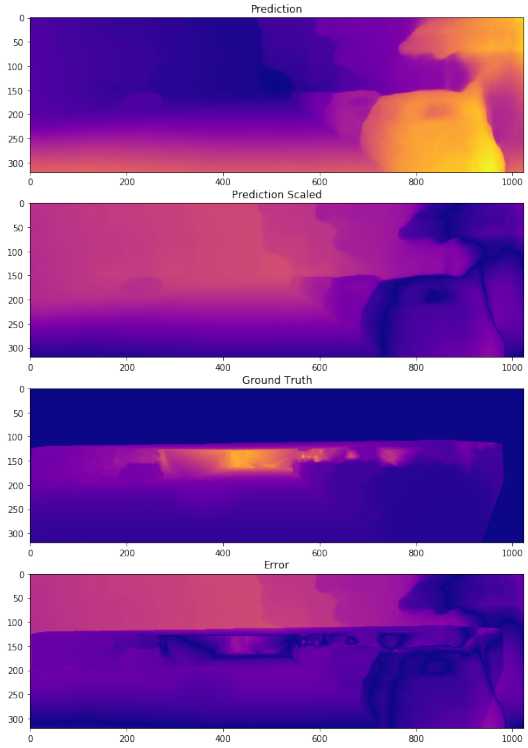

Fig. 29: Monodepth 2 Full Results for Image 100